2

USAFA-TR-91-17

CRITICAL TEMPERATURE DETERMINATION IN SUPERCONDUCTORS

C1C BRIAN A. DOYLE

CAPTAIN MATTHEW G. McHARG

DEPARTMENT OF PHYSICS
USAF ACADEMY, COLORADO 80840-5701

OCTOBER 1991

FINAL REPORT

DEAN OF THE FACULTY
UNITED STATES AIR FORCE ACADEMY
COLORADO 80840-5701

USAFA-TR-91-17

Technical Review by Capt Scott G. Wierschke
Department of Chemistry
USAFA Academy, Colorado 80840

Technical Review by Capt Brent A. Richert
Department of Physics
USAF Academy, Colorado 80840

Editorial Review by Lt Col Donald C. Anderson
Department of English
USAF Academy, Colorado 80840

This research report entitled "Critical Temperature Determination in Superconductors" is presented as a competent treatment of the subject, worthy of publication. The United States Air Force Academy vouches for the quality of the research, without necessarily endorsing the opinions and conclusions of the author.

This report has been cleared for open publication and public release by the appropriate Office of Information in accordance with AFM 190-1, AFR 12-30, and AFR 80-3. This report may have unlimited distribution.

Robert K. Morrow Jr.

ROBERT K. MORROW JR., Lt Col, USAF
Director of Research

31 Oct 91
Dated

REPORT DOCUMENTATION PAGE	Form Approved OMB No 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 18 June 1990	3. REPORT TYPE AND DATES COVERED Final

4. TITLE AND SUBTITLE Critical Temperature Determination In Superconductors	5. FUNDING NUMBERS
6. AUTHOR(S) Doyle, Brian A.; McHarg, Matthew G.	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Department of Physics United States Air Force Academy, CO 80840	8. PERFORMING ORGANIZATION REPORT NUMBER USAFA-TR-91-17
9. SPONSORING MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This project is based on a program that uses FORTRAN 77 to analyze resistance versus temperature data. The data are fit by two lines with different slopes and intercepts. A total chi-square for the fits is obtained. When the total chi-square is minimized, the critical temperature is assigned to the point where the data are broken into the intersecting two lines. The program also informs the user of the temperature at which those lines intersect. A sensitivity study was performed to find the relationship between uncertainties in the resistance and critical temperature. This analysis showed the resistance needed to be measured with less than 5% error to allow the Tc to be found with an error of three Kelvins.

14. SUBJECT TERMS Superconductor; Thin films; Critical Temperature; Sensitivity analysis	15. NUMBER OF PAGES 24
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT Unlimited

Critical Temperature Determination In Superconductors

C1C B. Doyle Capt M. McHarg

18 June 1990

Abstract

This project is based on a program that uses FORTRAN 77 to analyze resistance versus temperature data. The data are fit by two lines with different slopes and intercepts. A total χ^2 for the fits is obtained. When the total χ^2 is minimized, the critical temperature is assigned to the point where the data are broken into the intersecting two lines. The program also informs the user of the temperature at which those lines intersect. A sensitivity study was performed to find the relationship between uncertainties in the resistance and critical temperature. This analysis showed the resistance needed to be measured with less than 5% error to allow the T_c to be found with an error of three Kelvins.

1 Introduction

The recent discovery of high temperature superconductors has led to the study of how these superconductors work [1]. Of particular interest is the critical temperature (T_c) and slope of the resistance versus temperature graph below the critical temperature. This paper reports on a computer program that fits two lines to resistance versus temperature data from any superconductor. The program identifies the critical temperature and slope below the critical temperature of the superconductor.

Another program was made to provide sample data sets that have noise added to them. The noise was provided by a random number generator that sampled a gaussian distribution with zero mean and unit variance. The program output was used to study the relationship between the uncertainties in the resistance values and uncertainties in the critical temperature.

2 DESCRIPTION OF PROGRAM

This section describes the aspects of the programs, and details the theory behind the code. The listings have been labeled to facilitate following the description. The program that fits the lines (THIN.FOR) is addressed first.

In the main program, part 1, the variables are defined and the data and output files are opened. The files are briefly described in the comment block below the statements. The variables C and $CHECK$ are used in the least-squares subroutine and are addressed later. Part 2 counts the number of data points in the data set. It also asks the user if there is an error value for each data point. Part 3 calls the subroutine LSTSQR and checks to determine if another data break is needed.

Part 4 of the program is the subroutine LSTSQR (least-squares fit) [2]. This subroutine fits two lines with different slopes and intercepts to the data. It also determines how closely the lines fit the data by finding the total χ^2 of the two lines. The χ^2 is the residual value divided by the total standard deviation for all the points. The smaller χ^2, the closer the fitted line is to the actual data. This least- squares fit is repeated by continually moving the point at which the data are broken into the two lines back towards zero temperature. The T_c is assigned by the user when the χ^2 is minimized for both lines.

In Part 4a, two counters, LINENUM and COUNT, are initialized to one. LINENUM simply denotes which line is being fit, and COUNT is the current number of the data point being read in. The R and T data are read in from file RVST2.DAT. The temperature data is measured in Kelvins and the resistance data is a relative resistance with zero denoting no resistance and one denoting the maximum resistance for a given material. The data are arranged in two columns with the independent variable (temperature) on the left and the dependent variable (resistance) value on the right. The columns are separated by a comma. The values in RVST2.DAT can be integer or real. The data are read into two arrays X and Y. The values of X and Y are summed, and this information is used in determining the slope and intercept for the line that is being fit.

Part 4b removes the first data point in each set until a second point is input. This fills the requirement that a line needs two points to be defined. Part 4c determines the standard deviation of the data point. The standard deviation can be user defined, or can be approximated from the data by taking the square root of the data point. The standard deviation is later used to determine the χ^2. Part 4d checks to see if there are more data points to be fit. If not, the subroutine skips to the end. The slope and intercept are determined in 4e, and the values are written to the external file OUT.OUT in the format prescribed in line 55. Next, the residual value is computed in part 4f.

Part 4g is the test to determine if one line has been fit. Inside the subroutine, the program checks to see if a dummy variable, C (a counter), is greater than the predetermined number of checks desired. The number desired is set in the main program (CHECK). If there has not been enough data points run through the subroutine, then the program reads in another. If that is the point at which the user wants the first line break, then the subroutine assigns the temperature value to that of a variable called the critical temperature and begins to fit the second line. The same process is followed for the second line, and a new slope, intercept, and χ^2 are determined.

A total χ^2 is defined as the root mean sum of the individual χ^2 of the lines. This value is written to OUT.OUT, and the process is repeated with the break in the lines being moved back one data point from the previous run. The user seeks to minimize the total χ^2, and thus pick the proper place for the T_c. The program outputs the slope and intercept for both lines. These actions are incorporated in parts 4g, 4i, and 4j.

Part 4h is inserted to determine the errors associated with the resistance, the critical temperature, and the slope of the lines. All these values are output to different files as prescribed at the top of the main program. The error in the resistance is read from an external file (SIGMAR.DAT) as a percentage and then multiplied by the the value of R. Part 4K merely puts the output data in a form that is aesthetically pleasing.

2.1 NOISE

The second program, NOISE, generates random numbers based on the gaussian distribution of the data points and adds random "noise" to the data points [3]. These points can then be read by the first program as input. This allows THIN.FOR to be tested on data that is corrupted by noise of varying amounts.

In part 1 the data are read into the program exactly as in THIN.FOR. Part 2 either adds or subtracts "noise" from the resistance values. The choice of addition or subtraction is random. Part 3 randomly changes the seed from which the random number is generated.

There are two subroutines, URAND (part 5) and GAUSSN (part 4). URAND generates a random number between zero and one. It does this by a series of divisions and multiplications. The remainders of the divisions are used to produce the seed which is used with the intrinsic command FLOAT to produce a random number. This number is in turn used by GAUSSN to produce a normally distributed random number with unit variance. The program listings are found in Appendices A and B.

3 RESULTS

The sample data in RVST2.DAT, found in Appendix D, are characteristic of R versus T graphs [4]. The data have been hand digitized from the original paper which sustituted praseodyium into a bismuth strontium calcium copper oxide superconductor. The resistance versus temperature output in Appendix C dictates that the line be broken at approximately 104 deg K. This is very close to the middle of the "elbow" in the actual data between 115and 97K as seen graphically in Figure 1.

3.1 UNCERTAINTIES IN PARAMATERS

The uncertainty in the slope was determined as a function of the uncertainties in resistance. Following Bevington,

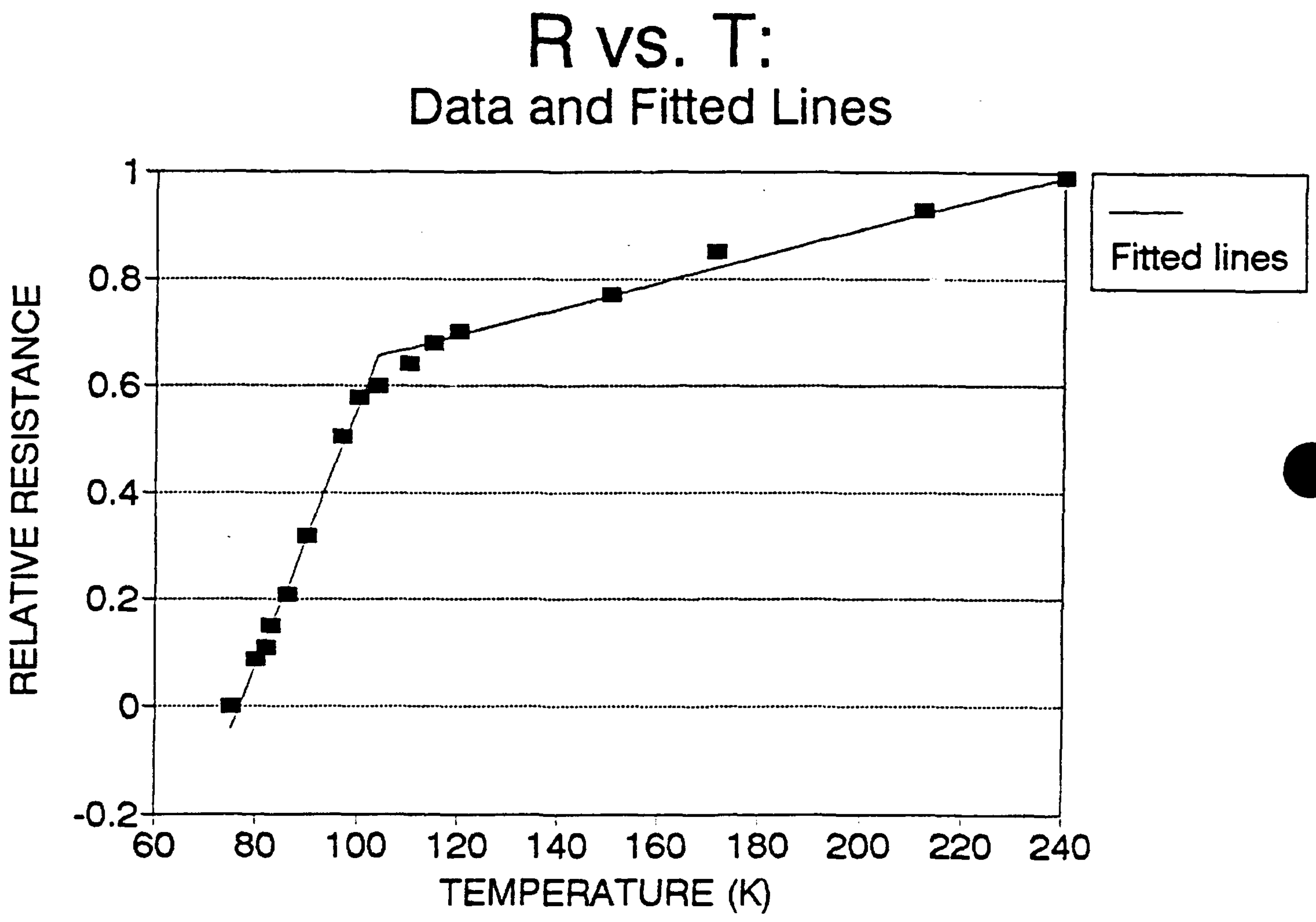

R vs. T:
Data and Fitted Lines
1
0.8
0.6
0.4
0.2
0
-0.2
60
80
100
120
140
160
180
200
220
240
TEMPERATURE (K)
RELATIVE RESISTANCE
Fitted lines

Figure 1

$$\sigma_b^2 = \sum_{j=1}^{N} \frac{\sigma_r^2}{\Delta^2}\left[N^2 x_j^2 - 2N x_j \sum x_i + (\sum x_i)^2\right]$$

where $\quad \Delta = N\sum x_i^2 - (\sum x_i)^2$

$$\sigma_b^2 = \frac{\sigma_r^2}{\Delta^2}\left[N^2\sum x_i^2) - 2N(\sum x_i)^2 + N(\sum x_i)^2\right] = \frac{N\sigma_r^2}{\Delta^2}\left[N\sum x_i^2 - (\sum x_i)^2\right] = \frac{N\sigma_r^2}{\Delta}$$

with $N = 16$, $\sigma_r = 0.003$, and $\Delta = 0.64$, we obtain $\frac{\sigma_b = 0.00079}{\mathrm{K}}$ [5].

The uncertainty in the critical temperature (T_c) was found by two methods. The first was found by using the slope of the first line and the σ_r at the point where the lines were broken (point slope method) The second was found by using the intersection of the two lines and incorporating in the calculation the errors in the slopes of both lines and their corresponding y-intercepts (intercept method). The equations for these methods are shown below.

Point Slope

$$\sigma_x = \sigma_{T_c} = \left[\left(\frac{\partial x}{\partial y}\right)^2 \sigma_y^2 + \left(\frac{\partial x}{\partial m}\right)^2 \sigma_m^2\right]^{\frac{1}{2}}$$

$$\sigma_{T_c} = \left[\left(\frac{1}{m}\right)^2 \sigma_y^2 + \left(\frac{y}{m^2}\right)^2 \sigma_m^2\right]^{\frac{1}{2}}$$

Intercept

$$\sigma_{T_c} = \left[\left(\frac{1}{m_1-m_2}\right)^2 \sigma_{b_2}^2 + \left(\frac{1}{m_1-m_2}\right)^2 \sigma_{b_1}^2 + \left(\frac{b_2-b_1}{(m_1-m_2)^2}\right)^2 \sigma_{m_1}^2 + \left(\frac{b_2-b_1}{(m_1-m_2)^2}\right)^2 \sigma_{m_2}^2\right]^{\frac{1}{2}}$$

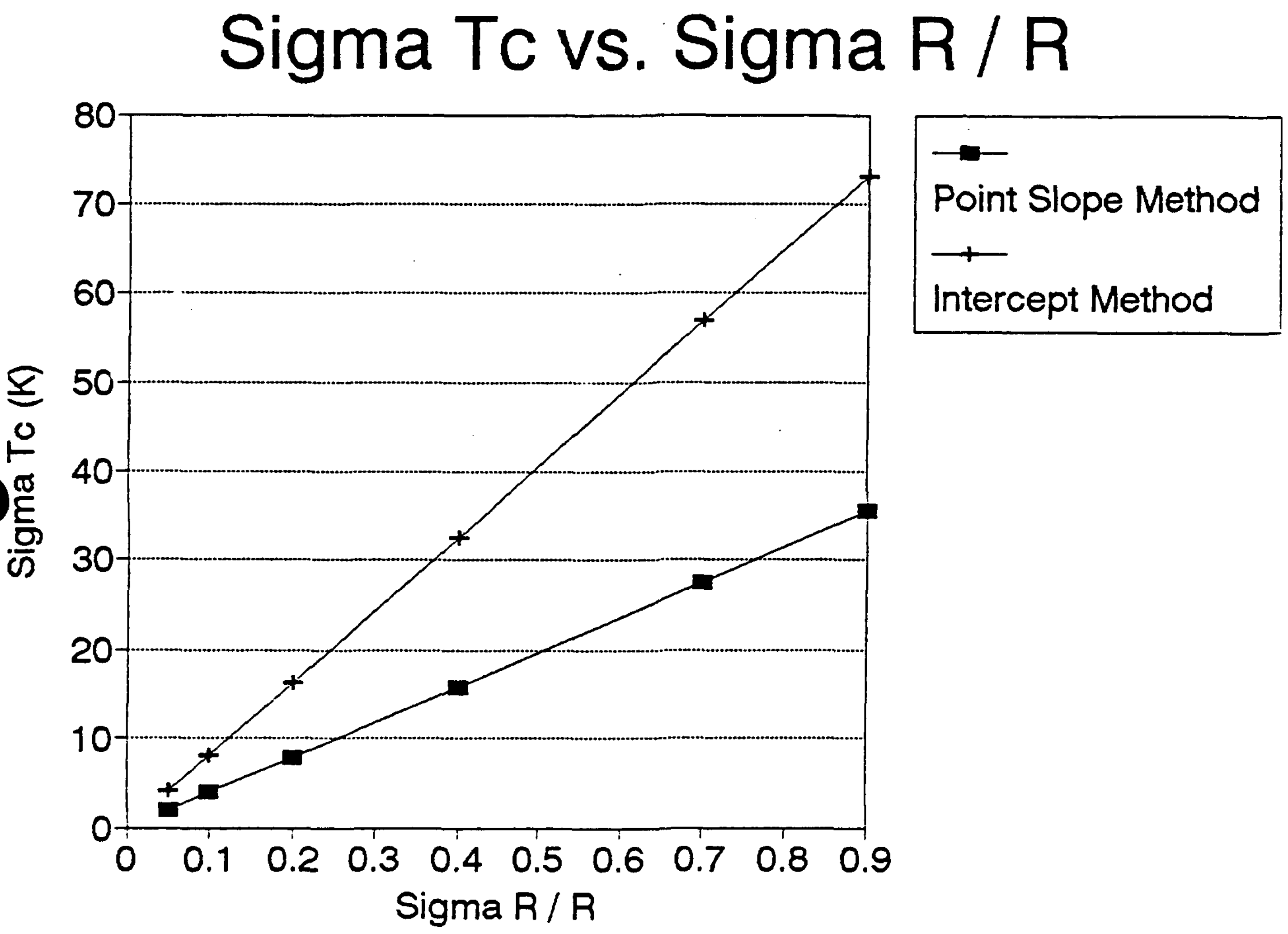
M is the respective slope for each line and b is the respective y intercept. Appendix C shows m and b for each fitted line.

TABLE 1

SIGMA R / R	SIGMA R @ TC	SIGMA TC (PT. SLOPE)	SIGMA TC (INTERSEPT)	SIGMA SLOPE
0.9	0.58	35.48 K	73.09 K	0.014214/K
0.7	0.45	27.60 K	56.86 K	0.011055/K
0.4	0.26	15.77 K	32.49 K	0.006317/K
0.2	0.13	7.88 K	16.23 K	0.003159/K
0.1	0.06	3.94 K	8.12 K	0.001580/K
0.05	0.03	1.97 K	4.06 K	0.000790/K

SIGMA SLOPE VS. SIGMA R	SLOPE OF LEFT LINE	TC
m = 0.02256/K	0.02/K	104K

SIGMA TC VS. SIGMA R
m = 58.43 K



Sigma Tc vs. Sigma R / R
80
70
60
50
40
30
20
10
0
Sigma Tc (K)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Sigma R / R
Point Slope Method
Intercept Method

Figure 2

Table 1 shows σ_{T_c} for different values of $\frac{\sigma_R}{R}$. Figure 2 shows the same data graphical y. Figure 2 shows a $\frac{\sigma_R}{R}$ of about 2 - 5 of approximately 3 K.

4 CONCLUSIONS

Using the hand digitized data in RVST2.DAT [4], the program THIN.FOR fits two lines to the resistance versus temperature data produced by thin film superconcuctors. The program determines the slope and intecepts of the two lines a id the critical temperature. The program also allows the user to determine where one line ends and the other starts, as appropriate with the user's neecs.

The uncertanty in the critical temperature was found to be related to the relative uncertainty in the resistance. It was found that a few percent relative measu ement of resistance is necessary to determine T_c with an error of 3 Kelvins. It was also found that the percent error in the slope is small compared to σ_R and σ_{T_c}.

5 REFERENCES

1. Chu, C.W., P.H. Hor, R.L. Meng, G. Gao, Z.J. Huang, and Y.Q. Wang, Phys. Rev. B *35*, 405 (1987).

2. Etter, D.M. *Structured Fortran 77 For Engineers and Scientists.* Menlo Park, California: Benjamin/Cummings Publishing Company, Inc., 1987.

3. Department of Astronautics, United States Air Force Academy. Personal interview. Jan. 1990.

4. Geller S., and K.Y. Wu. *Applied Physics Letters 54* (1989): 669.

5. Bevington, Phillip R. *Data Reduction and Error Analysis for the Physical Sciences.* New York: McGraw-Hill Book Company, 1969.

6 WORKS CONSULTED

Flannery, Brian P., William H.Press, Saul A. Teukolsky, William T. Vetterling. *Numerical Recipes*. New York: Cambridge.

APPENDIX A

```
*************************************************************************
*
*   TITLE:     THIN FILMS PROGRAM
*
*   AUTHOR:    BRIAN A. DOYLE, CS-10, x4500
*
*   DATE:      29 NOV 89
*
*   OVERVIEW:  This program analyzes resistance versus temperature.
*              It can be used in conjunction with superconductors.
*              A step by step analysis of the program can be found
*              in the "Description Of Program" section of the paper.
*
*
**************************************************************************
      PROGRAM THIN

      IMPLICIT LOGICAL (A-Z)
      INTEGER I,N, COUNT, NUMBERDAT, DEV
      REAL*8 X(500), Y(500), SLOPE, YINT, YNEW, RESID, SUMX, SUMY
      REAL*8 SUMXY, SUMXX, SUMRESID, CHITOT, CHECK, C
      DATA   COUNT, SUMX, SUMY, SUMXY, SUMXX, SUMRESID /1,5*0.0/

      OPEN (UNIT=10,FILE= 'RVST.DAT', STATUS = 'OLD')
      OPEN (UNIT=11,FILE= 'OUT.OUT', STATUS = 'NEW')
      OPEN (UNIT=12,FILE= 'FIT.DAT', STATUS = 'NEW')
      OPEN (UNIT=13,FILE= 'ERR.DAT', STATUS = 'OLD')                 1
      OPEN (UNIT=14,FILE= 'RVST2.DAT',STATUS = 'OLD')
      OPEN (UNIT=15,FILE= 'ERRSLOP.OUT',STATUS = 'NEW')
      OPEN (UNIT=16,FILE= 'ERRTC.OUT',STATUS = 'NEW')
      OPEN (UNIT=17,FILE= 'SIGMAR.DAT',STATUS = 'OLD')
*   10 = data points used to count
*   11 = output of thin
*   12 = data points of the fitted lines
*   13 = error for each data point
*   14 = data points used for least squares fit
*   15 = error associated with the slope
*   16 = error associated with the Tc
*   17 = percent error associated with R (it is altered)
      C = 0.1
      CHECK = 1.32

      DO 9 I = 1,1000
          READ (10,*) X(I)
        IF (X(I) .GE. 0) THEN
          NUMBERDAT = NUMBERDAT + 1
        ELSE                                                         2
          write(*,*) 'There are',numberdat,' data points.'
          CLOSE (10)
          GOTO 10
        ENDIF
9     CONTINUE

    Switch to enter error or use square root of data point *
10    WRITE(*,*) 'Input (1) if you have an error value for each data poi
     &nt.'
      WRITE(*,*) 'Otherwise, enter (2)'
      READ (*,*) DEV

*   Go thru process of fitting lines until there are no more data points
      I = 0
      DO 8 I = 1, NUMBERDAT
1     CALL LSTSQR(X, Y, NUMBERDAT, CHITOT, CHECK, C, COUNT, DEV)
        IF (CHITOT .GT. C) THEN
          CHECK = CHECK -.091                                        3
          C = CHITOT
          COUNT = 1
        ENDIF
8     CONTINUE
*   Closing the files previously opened.
      CLOSE (17)
      CLOSE (16)
      CLOSE (15)
      CLOSE (14)
      CLOSE (13)
      CLOSE (12)
      CLOSE (11)
      CLOSE (10)
      STOP 'OUTSTANDING'
      END

************************************************************************
*
*     TITLE:  LSTSQR
*    AUTHOR:  BRIAN A. DOYLE, CS-10, x4500
*      DATE:  29 NOV 89
*  OVERVIEW:  This subroutine uses the least squares fit method to fit
*             the R and T data, in an external file, to a line.
*   SOURCES:  See pages 382-388, 453 in Structured Fortran by Etter.
************************************************************************
      SUBROUTINE LSTSQR(X,Y,NUMBERDAT,CHITOT,CHECK,C,COUNT,DEV)

      IMPLICIT LOGICAL (A-Z)
      INTEGER COUNT, N, I, LINENUM, NUMDATA, NUMBERDAT, DEV, RUFF, ARF
      INTEGER GOODFIT, Q
      REAL*8 X(500), Y(500), SLOPE, YINT, YNEW, RESID, SUMX, SUMY
      REAL*8 SUMXY, SUMXX, SUMRESID, CHI1, SIGMA(500), SIGMATOT, CHITOT
      REAL*8 CHI2, CHECK, C, SIGTC, SIGSLOPE, ERRY, SIG, DELTA, SUMX2
      REAL*8 SLO1, SLO2, YINT1, YINT2, INTSEC

      GOODFIT = 0
      LINENUM = 1
      COUNT = 1
**    Reading in data points. **
**    This is so I can mess with NUMDATA and still know how many
*     points there are.
      NUMDATA = NUMBERDAT

5     READ (14,*) X(COUNT), Y(COUNT)
         SUMX = SUMX + X(COUNT)
         SUMY = SUMY + Y(COUNT)                                     4a
         SUMXY = SUMXY + X(COUNT) * Y(COUNT)
         SUMXX = SUMXX + X(COUNT) * X(COUNT)

**    This is to throw out the first data point of each line because
*     it produces bad results.
         IF (COUNT .EQ. 1) THEN
```

```
          COUNT = COUNT + 1
          GOTO 5                                                4b
        ENDIF

        COUNT = COUNT + 1
        N = COUNT - 1
**    Figuring the standard deviation for each point. **
*   The SD is either read in or the square root is used.
      IF (DEV .EQ. 1) THEN
        READ (13,*) SIGMA(N)
      ELSE
        SIGMA(N) = DSQRT(X(N))
        SIGMATOT = SIGMA(N) + SIGMATOT                          4c
      ENDIF
*   If there are no more data points, go to the end.
        IF (COUNT .GT. NUMDATA) THEN
          GOTO 53
        ENDIF                                                   4d
        IF (N .LE. 1) THEN
          N = 2
        ENDIF

**    Determining slope and y intercept of one line. **
      SLOPE = (SUMX * SUMY - REAL(N) * SUMXY) /
     &        (SUMX * SUMX - REAL(N) * SUMXX)
      YINT = (SUMY - SLOPE * SUMX) / REAL(N)

      IF (LINENUM .EQ. 1) THEN
          SLO1 = SLOPE
          YINT1 = YINT                                          4e
        ELSE
          SLO2 = SLOPE
          YINT2 = YINT
      ENDIF
      WRITE(11,*) 'THE LINEAR EQUATION IS FOR LINE #',LINENUM
      WRITE(11,55) SLOPE, YINT

**    Calculating a predicted value for y and determining how good the
*     fit is, based on this value.
      DO 65 I = 1,N
        YNEW = SLOPE * X(I) + YINT
          IF (C .GE. CHECK)  THEN
            WRITE (12,91) X(I), YNEW
          ENDIF
        RUFF = ARF + I - 1.0D0
          IF (LINENUM .EQ. 2 .AND. COUNT .EQ. NUMDATA) THEN
            WRITE (12,92) X(RUFF), YNEW                         4f
          ENDIF
        RESID = Y(I) - YNEW
        SUMRESID = SUMRESID + RESID * RESID
*        WRITE(11,60) X(I), Y(I), YNEW, RESID
65    CONTINUE

*      WRITE (11,*)
*      WRITE (11,70) SUMRESID

     Checking to see if the fit is good for one line or if it is
*    trying to use a point on the second line.  If it is not good,
*    the fit for a second line is started.
      IF (C .LT. CHECK) THEN
```

```
          C = C + .1
          GOTO 5
        ELSE
*    When one line has been fit, the following happens.
          C = -5.0
          WRITE(*,*) 'AT DATA POINT NUMBER ',N,', ONE LINE HAS BEEN FIT.'
          WRITE(11,75) X(N)
          CHI1 = DSQRT((SUMRESID / SIGMATOT) ** 2)                    4g
          CHI1 = CHI1 * 100.0D0
          WRITE (11,80) CHI1
          CHI1 = CHI1 / 100.0D0
*    For the best fit (goodfit), the errors are written to another file.
          GOODFIT = 10
          IF (N .EQ. GOODFIT) THEN
              READ (17,*) ERRY
              Q = N - 2
        write(*,*) 'y(Q) = ',y(Q)
              SIG = ERRY * Y(Q)                                       4h
              SIGTC = SIG / SLOPE
*    This comes from page 114 in Bevington
              SUMX2 = SUMX ** 2
              DELTA = (N * SUMXX) - SUMX2
              SIGSLOPE = DSQRT(N * (SIG ** 2) / DELTA)
              WRITE(15,93) SIG, SIGSLOPE
              WRITE(16,91) SIG, SIGTC
          ENDIF

          NUMDATA = NUMDATA - N
          SUMX = 0.0D0
          SUMY = 0.0D0
          SUMXY = 0.0D0                                               4i
          SUMXX = 0.0D0
          SUMRESID = 0.0D0
          LINENUM = 2
          ARF = COUNT
          COUNT = 1
          GOTO 5
        ENDIF

*    Calculating chi for the 2nd line and the total chi squared.
53      IF (LINENUM .EQ. 2) THEN
          CHI2 = DSQRT((SUMRESID / SIGMATOT) ** 2)
          CHITOT = DSQRT(CHI1 ** 2 + CHI2 ** 2)
          CHI2 = CHI2 * 100.0D0
          WRITE (11,85) CHI2
          CHI2 = CHI2 / 100.0D0
          WRITE (11,87)
          CHITOT = CHITOT * 100.0D0
          WRITE (11,88) CHITOT
          CHITOT = CHITOT / 100.0D0
          WRITE (11,89)                                               4j
        ENDIF
          SUMX = 0.0D0
          SUMY = 0.0D0
          SUMXY = 0.0D0
          SUMXX = 0.0D0
          SUMRESID = 0.0D0
        WRITE (12,*) '*********'
          INTSEC = (YINT2 - YINT1) / (SLO1 - SLO2)
*        WRITE (*,*) 'THE INTERSECTION IS AT ',INTSEC,' K'
```

```
54     RETURN
*   This is only to make it look nice.
55     FORMAT (1X,'Y = ',F6.2,' X + ',F6.2)
60     FORMAT (1X,F6.2,6X,F6.2,6X,F6.2,7X,F6.2)
70     FORMAT (1X, 'RESIDUAL SUM =', F6.2)
75     FORMAT (1X, 'THE CRITICAL TEMPERATURE IS ', F6.2 ,' DEGREES K')
80     FORMAT (1X, 'THE VALUE OF CHI FOR 1ST LINE: ', F7.4, /)
85     FORMAT (1X, 'THE VALUE OF CHI FOR 2ND LINE: ', F7.4)
87     FORMAT (1X,/,'******************************')                4k
88     FORMAT (1X,  'TOTAL CHI SQUARED FOR BOTH LINES: ', F7.4)
89     FORMAT (1X,  '******************************',/)
91     FORMAT (5X, F6.2, 3X, F6.2)
92     FORMAT (5X, F6.2, 3X, F6.2)
93     FORMAT (5X, F6.2, 3X, F8.6)

       END
```

APPENDIX B

```
***********************************************************************
*
*
*      TITLE:       RANDOM NOISE GENERATOR
*
*      AUTHOR:      DEPARTMENT OF ASTRONAUTICS, USAFA
*
*      DATE:        16 JAN 90
*
*      OVERVIEW:    This program generates random numbers by sampling
*                   a gaussian distribution with zero mean and unit
*                   variance.  The program adds or substracts values
*                   from R vs. T data to create "noise" for a sensitivity
*                   check.
***********************************************************************

      PROGRAM NOISE

      REAL*8 YNOISE(500),X(500),Y(500),TEMP,TEMP2
      INTEGER NUMBERDAT,SEED,I,COUNT, ADDOR

      OPEN (UNIT = 5,FILE='RVST.DAT',STATUS = 'OLD')
      OPEN (UNIT = 6, FILE = 'RAND.DAT', STATUS = 'NEW')

      ADDOR = 1
      NUMBERDAT = 1
      DO 9 I = 1,1000
          READ (5,*) X(I)
        IF (X(I) .GE. 0) THEN
          NUMBERDAT = NUMBERDAT + 1
        ELSE                                                       1
          NUMBERDAT = NUMBERDAT - 1
          CLOSE (5)
          GOTO 10
        ENDIF
9     CONTINUE

10    OPEN (UNIT=5, FILE = 'RVST.DAT', STATUS = 'OLD')
        SEED = 824064364
      DO 11 COUNT = 1,NUMBERDAT
        READ (5,*) X(COUNT), Y(COUNT)
        SIG = Y(COUNT)
        TEMP = URAND(SEED)
        TEMP2 = GAUSSN(SIG, SEED) / 100

        IF (ADDOR .EQ. 2) THEN
            Y(COUNT) = Y(COUNT) + TEMP2                            2
        ELSE
            Y(COUNT) = Y(COUNT) - TEMP2
        ENDIF
        WRITE (6,50) X(COUNT), Y(COUNT)
        IF (TEMP .GT. .5) THEN
          ADDOR = 2
          SEED = SEED + (1000 * TEMP)
        ELSE                                                       3
          SEED = SEED - (1000 * TEMP)
        ENDIF
11    CONTINUE
```

```
      CLOSE(5)
      CLOSE(6)
50    FORMAT(10X, F7.2, F7.2)
      STOP 'DONE'
      END

      REAL FUNCTION GAUSSN(SIG, SEED)

      IMPLICIT LOGICAL (A-Z)
      REAL*8 GNOIZ, SIG
      INTEGER*4 SEED, I

      SIG = 5.0
      GNOIZ = 0.
      DO 10 I = 1,12                                                  4
          GNOIZ = GNOIZ + URAND(SEED)
10    CONTINUE
      GAUSSN = SIG * (GNOIZ - 6.0)
      RETURN
      END

      REAL FUNCTION URAND(SEED)

      IMPLICIT LOGICAL (A-Z)
      INTEGER*4 B2E15,B2E16,MODLUS,HIGH15,HIGH31,LOW15,LOWPRD
      INTEGER*4 MULT1,MULT2,OVFLOW,SEED
      DATA MULT1,MULT2/24112,26143/
      DATA B2E15,B2E16,MODLUS/32768,65536,2147483647/

      HIGH15 = SEED/B2E16
      LOWPRD = (SEED - HIGH15*B2E16) * MULT1
      LOW15 = LOWPRD/B2E16
      HIGH31 = HIGH15 * MULT1 + LOW15
      OVFLOW = HIGH31/B2E15

      SEED = (((LOWPRD - LOW15 * B2E16) - MODLUS) +                   5
     &          (HIGH31 - OVFLOW * B2E15) * B2E16) + OVFLOW
      IF (SEED .LT. 0) SEED = SEED + MODLUS

      HIGH15 = SEED/B2E16
      LOWPRD = (SEED - HIGH15 * B2E16) * MULT2
      LOW15 = LOWPRD / B2E16
      HIGH31 = HIGH15 * MULT2 + LOW15
      OVFLOW = HIGH31 / B2E15

      SEED = (((LOWPRD - LOW15 * B2E16) - MODLUS) +
     &          (HIGH31 - OVFLOW * B2E15) * B2E16) + OVFLOW
      IF (SEED.LT.0) SEED = SEED + MODLUS

      URAND = FLOAT(2*(SEED/256) + 1)/16777216.0
      RETURN
      END
```

APPENDIX C

This appendix contains the output of the program THIN.FOR for the resistance versus temperature data contained in Appendix D. The critical value is the total chi squared for both lines which is denoted by two rows of stars. This value informs the user of the best least-squares fit match and its associated critical temperature.

```
THE LINEAR EQUATION IS FOR LINE #            1
Y =     .00 X +    -.03
THE CRITICAL TEMPERATURE IS 240.00 DEGREES K
THE VALUE OF CHI FOR 1ST LINE:    .2083

THE VALUE OF CHI FOR 2ND LINE:    .0000

*****************************
TOTAL CHI SQUARED FOR BOTH LINES:    .2083
*****************************

THE LINEAR EQUATION IS FOR LINE #            1
Y =     .01 X +    -.12
THE CRITICAL TEMPERATURE IS 212.00 DEGREES K
THE VALUE OF CHI FOR 1ST LINE:    .1136

THE LINEAR EQUATION IS FOR LINE #            2
Y =     .01 X +    -.45
THE VALUE OF CHI FOR 2ND LINE:    .0000

*****************************
TOTAL CHI SQUARED FOR BOTH LINES:    .1136
*****************************

THE LINEAR EQUATION IS FOR LINE #            1
Y =     .01 X +    -.27
THE CRITICAL TEMPERATURE IS 181.00 DEGREES K
THE VALUE OF CHI FOR 1ST LINE:    .0589

THE LINEAR EQUATION IS FOR LINE #            2
Y =     .01 X +    -.46
THE VALUE OF CHI FOR 2ND LINE:    .0009

*****************************
TOTAL CHI SQUARED FOR BOTH LINES:    .0589
*****************************

THE LINEAR EQUATION IS FOR LINE #            1
Y =     .01 X +    -.59
THE CRITICAL TEMPERATURE IS 150.00 DEGREES K
THE VALUE OF CHI FOR 1ST LINE:    .0258

THE LINEAR EQUATION IS FOR LINE #            2
Y =     .01 X +    -.42
THE VALUE OF CHI FOR 2ND LINE:    .0033

*****************************
TOTAL CHI SQUARED FOR BOTH LINES:    .0260
*****************************

THE LINEAR EQUATION IS FOR LINE #            1
Y =     .02 X +   -1.15
THE CRITICAL TEMPERATURE IS 120.00 DEGREES K
THE VALUE OF CHI FOR 1ST LINE:    .0087

THE LINEAR EQUATION IS FOR LINE #            2
Y =     .01 X +    -.32
THE VALUE OF CHI FOR 2ND LINE:    .0072

*****************************
TOTAL CHI SQUARED FOR BOTH LINES:    .0113
*****************************

THE LINEAR EQUATION IS FOR LINE #            1
Y =   [illegible]02 X +   -1.33
THE [illegible]ICAL TEMPERATURE IS 115.00 DEGREES K
THE [illegible]E OF CHI FOR 1ST LINE:   .0040

THE [illegible]AR EQUATION IS FOR LINE #          2
Y =    .01 X +   -.15
THE VALUE OF CHI FOR 2ND LINE:   .0117

****************************
 TOTAL CHI SQUARED FOR BOTH LINES:   .0124
 ****************************

 THE LINEAR EQUATION IS FOR LINE #          1
 Y =    .02 X +  -1.54
 THE CRITICAL TEMPERATURE IS 110.00 DEGREES K
 THE VALUE OF CHI FOR 1ST LINE:   .0014

 THE LINEAR EQUATION IS FOR LINE #          2
 Y =    .00 X +   -.05
 THE VALUE OF CHI FOR 2ND LINE:   .0120

****************************
 TOTAL CHI SQUARED FOR BOTH LINES:   .0120
 ****************************

 THE LINEAR EQUATION IS FOR LINE #          1
 Y =    .02 X +  -1.82
 THE CRITICAL TEMPERATURE IS 104.00 DEGREES K
 THE VALUE OF CHI FOR 1ST LINE:   .0003

 THE LINEAR EQUATION IS FOR LINE #          2
 Y =    .00 X +    .02
 THE VALUE OF CHI FOR 2ND LINE:   .0122

****************************
 TOTAL CHI SQUARED FOR BOTH LINES:   .0122
 ****************************

 THE LINEAR EQUATION IS FOR LINE #          1
 Y =    .03 X +  -2.01
 THE CRITICAL TEMPERATURE IS 100.00 DEGREES K
 THE VALUE OF CHI FOR 1ST LINE:   .0000

 THE LINEAR EQUATION IS FOR LINE #          2
 Y =    .00 X +    .06
 THE VALUE OF CHI FOR 2ND LINE:   .0121

****************************
 TOTAL CHI SQUARED FOR BOTH LINES:   .0121
 ****************************

 THE LINEAR EQUATION IS FOR LINE #          1
 Y =    .03 X +  -2.01
 THE CRITICAL TEMPERATURE IS  97.00 DEGREES K
 THE VALUE OF CHI FOR 1ST LINE:   .0000

 THE LINEAR EQUATION IS FOR LINE #          2
 Y =    .00 X +    .09
 THE VALUE OF CHI FOR 2ND LINE:   .0117

****************************
 TOTAL'CHI SQUARED FOR BOTH LINES:   .0117
 ****************************

 THE LINEAR EQUATION IS FOR LINE #          1
 Y =    .03 X +  -1.96
 THE CRITICAL TEMPERATURE IS  90.00 DEGREES K
 THE VALUE OF CHI FOR 1ST LINE:   .0000
 THE LINEAR EQUATION IS FOR LINE #            2
 Y =     .00 X +     .09
 THE VALUE OF CHI FOR 2ND LINE:    .0124

*****************************
 TOTAL CHI SQUARED FOR BOTH LINES:    .0124
 *****************************

 THE LINEAR EQUATION IS FOR LINE #            1
 Y =     .02 X +   -1.84
 THE CRITICAL TEMPERATURE IS  86.00 DEGREES K
 THE VALUE OF CHI FOR 1ST LINE:    .0000

 THE LINEAR EQUATION IS FOR LINE #            2
 Y =     .00 X +     .05
 THE VALUE OF CHI FOR 2ND LINE:    .0210

*****************************
 TOTAL CHI SQUARED FOR BOTH LINES:    .0210
 *****************************

 THE LINEAR EQUATION IS FOR LINE #            1
 Y =     .04 X +   -3.17
 THE CRITICAL TEMPERATURE IS  83.00 DEGREES K
 THE VALUE OF CHI FOR 1ST LINE:    .0000

 THE LINEAR EQUATION IS FOR LINE #            2
 Y =     .00 X +     .00
 THE VALUE OF CHI FOR 2ND LINE:    .0321

*****************************
 TOTAL CHI SQUARED FOR BOTH LINES:    .0321
 *****************************
```

APPENDIX D

This appendix contains the resistance versus temperature data for the output contained in Appendix C. The data was hand-digitized from a previously published source [4].

TEMPERATURE (K)	RESISTANCE
75	.00
80	.09
82	.11
83	.15
86	.21
90	.32
97	.505
100	.58
104	.60
110	.64
115	.68
120	.70
150	.77
181	.85
212	.93
240	.99